

# Label Smoothing++: Enhanced Label Regularization for Training Neural Networks

Sachin Chhabra<sup>1</sup>  
sachin.chhabra@asu.edu

Hemanth Venkateswara<sup>2</sup>  
hvenkateswara@gsu.edu

Baoxin Li<sup>1</sup>  
baoxin.li@asu.edu

<sup>1</sup> Arizona State University  
699 S Mill Ave.  
Tempe, AZ, USA- 85281

<sup>2</sup> Georgia State University  
25 Park PI NE  
Atlanta, GA, USA - 30303

---

## Abstract

Training neural networks with one-hot target labels often results in overconfidence and overfitting. Label smoothing addresses this issue by perturbing the one-hot target labels by adding a uniform probability vector to create a regularized label. Although label smoothing improves the network’s generalization ability, it assigns equal importance to all the non-target classes, which destroys the inter-class relationships. In this paper, we propose a novel label regularization training strategy called Label Smoothing++, which assigns non-zero probabilities to non-target classes and accounts for their inter-class relationships. Our approach uses a fixed label for the target class while enabling the network to learn the labels associated with non-target classes. Through extensive experiments on multiple datasets, we demonstrate how Label Smoothing++ mitigates overconfident predictions while promoting inter-class relationships and generalization capabilities.

## 1 Introduction

One of the most common practices for training neural networks is cross-entropy loss with one-hot target labels. However, it has been shown that this leads to overfitting and overconfident predictions by the network [29]. Numerous regularization techniques have been proposed to impose additional constraints to tackle this issue. Some of these techniques like Cutout [6], Mixup [38], CutMix [36], and others [0, 8, 10] alter the input data and are applied without considering the object positions, potentially impacting such entities directly. An alternative approach is label regularization, which operates on training labels. Label smoothing is one of the easiest methods that create regularized targets by taking a weighted sum of the one-hot probability vector and a uniform vector based on a hyperparameter  $\alpha$  to mitigate the overconfidence problem.

Nowadays, Label Smoothing has become one of the standard ways of training neural networks [0, 19]. Even though it provides benefits in the form of generalization, it is known to eliminate inter-class relationships [23]. By using a uniform probability vector, Label Smoothing assigns equal weight to all the non-target classes, which means all the classes are equally different from the target class. However, this is not always the case. For example,

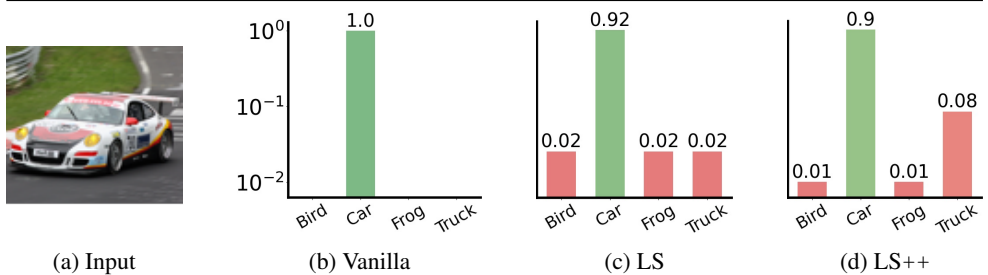


Figure 1: Different types of labels generated by various sources. (a) Input. (b) Traditionally (Vanilla), 1-hot probability vectors are used as labels. (c) Label Smoothing (LS) evenly distributes a probability parameter (denoted as  $\alpha$ , with a value of 0.1 in this context) across all classes. (d) Label Smoothing (LS++) distributes  $\alpha$  probability among the non-target classes for all classes independently.

consider a 4-way classification of *Bird*, *Car*, *Frog*, and *Truck*. Here, the training label for the *Car* class should have more weight assigned to *Truck* given their semantic similarity as compared to other classes like *Frog* and *Bird*. Such inter-class relationships, overlooked by Label Smoothing, can help enhance generalization abilities, knowledge distillation, learning from noisy labels, and handling missing data [11, 23, 57].

This paper proposes a novel label regularization method called Label Smoothing++ (LS++) that generates regularized training labels from one-hot target labels. It is done by retaining the confidence of the target class to be high while also assigning non-zero probabilities to the non-target classes by accounting for inter-class relationships. In standard Label Smoothing, the 1-hot target label is perturbed by adding a uniform distribution to the 1-hot target vector. In Label Smoothing++, we determine a class-wise probability vector to add to the 1-hot vector. Here, the samples of a class are constrained to produce the same outputs for all the classes.

Label Smoothing++ provides flexibility in how the probabilities are assigned among the non-target classes, which is essential for learning inter-class relationships. Refer to Figure 1 where we display targets generated by different label regularization techniques for a 4-class classification problem. Through experiments on multiple datasets and in various settings, we show the strengths of our proposed method, Label Smoothing++, compared to other label regularization techniques.

## 2 Related Work

Using a 1-hot target label in neural network training is known to lead to overconfidence and hinder generalization [29]. Label regularization techniques aim to modify training labels to mitigate overconfidence. One of the earliest and easiest label regularization methods is Label Smoothing, which combines the 1-hot vector with a uniform vector based on a hyperparameter  $\alpha$  [29]. Despite its advantages, Label Smoothing destroys inter-class relationships by assigning equal weights to all non-target classes [23]. We aim to deviate from using a uniform vector by offering the network the flexibility to adjust its training label.

An alternative for regularizing network predictions is entropy maximization [26]. Entropy maximization directly penalizes the network for overconfident predictions. This tech-

nique provides greater flexibility but requires hyperparameter tuning for the entropy maximization loss weight. Focal loss is a modification of the cross-entropy loss function that was introduced to address overconfidence by assigning higher weights to samples with low confidence and lower weights to those with high confidence [20, 22]. This approach minimizes entropy maximization and a regularized KL divergence to prevent the network from becoming excessively overconfident.

Knowledge distillation is considered a form of label regularization that involves generating targets from a larger network (the Teacher) and transferring this knowledge to a smaller network (the Student) on a per-sample basis [11, 35]. The relationship of each sample to non-target classes, as learned by the Teacher, helps regulate the student networks [11]. In alignment with this concept, a trained network was used to train another network with the same architecture in Teacher-Free Knowledge Distillation [35]. However, this approach incurs significant computational expenses as it requires training a network twice and generating outputs online. An alternative, Teacher-Free regularization, behaves similarly to Label Smoothing but utilizes a high mixing coefficient of 0.9 to generate a smoothed probability vector [35]. The network is trained to align predicted probabilities with this vector at a high temperature, reducing computational costs but still relying on a uniform vector.

Online Label Smoothing is another label regularization approach that is based on network predictions [35]. It computes average network predictions for each class and mixes them with a 1-hot probability vector. While it diminishes the need to train the network twice, it carries a substantial computational overhead as average network predictions must be computed every epoch on the training set. Our approach also has a class-based alignment (without the computational overhead of computing it every epoch) but only allows changes in the distribution of probabilities among the non-target classes, unlike online Label Smoothing, where training labels become 1-hot when network predictions tend towards 1-hot.

## 3 Methodology

### 3.1 Background

Consider a dataset  $D := \{(x_i, y_i)\}_{i=1}^m$  with  $K$  classes. For a pair  $(x, y)$ ,  $x$  is the input and  $y$  is its corresponding target with  $y \in \{1, 2, \dots, K\}$ . Let  $\bar{y} = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_K]^\top$  denote the one-hot presentation of the target label  $y$ . A neural network  $G$  takes  $x$  as input and generates a probability vector  $G(x) = \hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_K]$ . Here,  $\hat{y}$  is the predicted probability for the input  $x$ . The traditional procedure for training  $G$  is minimizing the cross-entropy loss,

$$H(\bar{y}, \hat{y}) = -\bar{y} \log \hat{y} = -\sum_{i=1}^K \bar{y}_i \log \hat{y}_i = -\log \hat{y}_y. \quad (1)$$

Training a neural network with 1-hot target labels often leads to issues of overconfidence and overfitting [29]. Label smoothing is a popular label regularization technique that alleviates this problem. It modifies the training label by using a weighted combination of the 1-hot and a uniform probability vector.

$$\bar{y}^{ls} = (1 - \alpha)\bar{y} + \alpha u, \quad (2)$$

where  $u = [\frac{1}{K}, \dots, \frac{1}{K}]^\top$  is a uniform probability vector of size  $K$  and each element is equal to  $\frac{1}{K}$ .  $\alpha$  is a hyperparameter that decides the weight between 1-hot and the uniform probability vector. Label Smoothing trains the network minimizing the same cross-entropy loss but with regularized training label:  $H(\bar{y}^{ls}, \hat{y}) = -\bar{y}^{ls} \log \hat{y} = -\sum_{i=1}^K \bar{y}_i^{ls} \log \hat{y}_i$ .

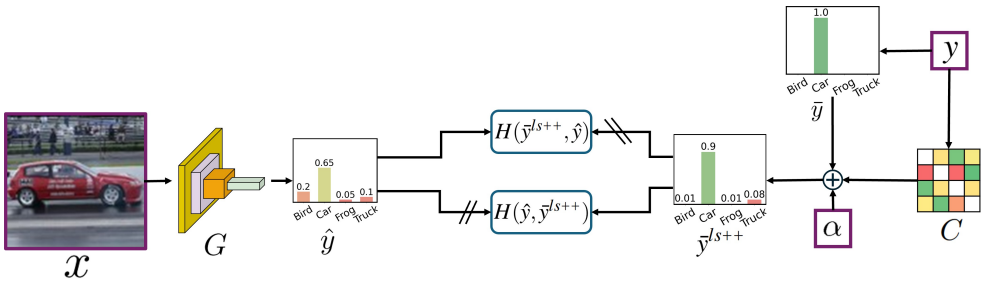


Figure 2: Model Diagram of Label Smoothing++ (LS++). Our approach distributes  $\alpha$  confidence among the non-target classes using learnable targets for each class independently. This promotes all samples of a class to achieve similar output. We train Label Smoothing++ with symmetric cross-entropy loss but because of learnable targets, we stop the flow of gradients from loss to some parameters (visually represented by double-slant lines).

### 3.2 Label Smoothing++

The idea behind label regularization is to reduce the confidence of the target class by a value  $\alpha$  and increment the confidence of all classes by the same amount  $\alpha$ . Label Smoothing distributes the value  $\alpha$  uniformly among all the classes. Instead of using a uniform assignment, we propose to learn the optimal assignment. We assume that the samples of a class generally share some similar characteristics. So, their output vector should share similarities as well. With this understanding, we propose Label Smoothing++ (LS++).

We train the network to learn the assignment of the residual probability  $\alpha$  among the non-target classes for each target class independently. For every class  $y$  we learn a probability vector  $C_y$  of length  $(K - 1)$  which represents the probability assignment of the non-target  $K - 1$  classes. We used a  $K - 1$  length to distribute the residual  $\alpha$  probability among the non-target classes only. This ensures the probability of the target class remains unchanged and only the non-target classes are adjusted. The regularized training label is then given by,

$$\hat{y}^{LS++} = (1 - \alpha)\hat{y} + \alpha C_y. \quad (3)$$

Note:  $C_y$  has size  $K - 1$  but is adjusted to be length  $K$  after inserting a 0 at the ground truth position  $y$  in eq. 3. We train the network to predict the same regularized training label for all samples belonging to a class. The  $\{C_y\}_{y=1}^K$  vectors together form the  $C$  matrix of dimension  $K \times K$  which is estimated by training. The  $C$ -Matrix has a diagonal element set to 0. Label Smoothing++ provides the same mixing training label for samples belonging to a class but different for each class. Our goal here is to provide freedom to the network to choose its optimal training label while adhering to the class-level constraint. Since the label for the target class is fixed, we only need to impose consistency on the non-target classes. Note: The  $C$ -Matrix is not a symmetrical matrix, as class relationships can differ based on the query class. We discuss this in more detail in the supplementary material.

Label Smoothing++ has training labels  $\hat{y}^{LS++}$  that need to be learned. For training the network, we depart from the traditional cross-entropy loss, which works well only for fixed training labels. The cross-entropy loss is an upper-bound on the Kullback-Leibler (KL) divergence with,  $H(\hat{y}^{LS++}, \hat{y}) = KL(\hat{y}^{LS++} || \hat{y}) + H(\hat{y}^{LS++})$ , where  $H(\hat{y}^{LS++})$  is the entropy of the training label  $\hat{y}^{LS++}$ . When the training label is fixed, such as in 1-hot or Label Smoothing, the entropy is merely a constant. But with  $\hat{y}^{LS++}$ , the training label contains learnable parameter

Method	ResNet18	ResNet34	ResNet50	ResNet101	DenseNet121
1-hot	75.87	79.38	78.79	79.66	79.04
LS [24]	77.26	79.06	78.80	79.88	80.38
TFKD <sub>self</sub> [65]	77.10	-	-	-	80.26
TFKD <sub>reg</sub> [65]	77.36	-	-	-	-
OLS [64]	-	79.96	79.35	80.34	-
Zipf's LS [68]	77.38	-	-	-	79.03
FL-3 [24]	-	-	77.25	-	-
FLSD-53 [24]	-	-	76.78	-	-
LS++ (Ours)	<b>79.33±0.23</b>	<b>80.25±0.14</b>	<b>81.05±0.73</b>	<b>81.13±0.52</b>	<b>80.71±0.13</b>

Table 1: Top-1 Classification accuracy on CIFAR100 dataset using different networks.

C. Applying an entropy minimization loss on  $\bar{y}^{ls++}$  results in assigning all the probability to one of the classes in  $C_y$ , which is undesirable. We address this using a symmetric cross-entropy loss  $H(\bar{y}^{ls++}, \hat{y}) + H(\hat{y}, \bar{y}^{ls++})$  where the network parameters  $G$  are trained using the first term  $H(\bar{y}^{ls++}, \hat{y})$ , and the  $C$  matrix is trained using the second term  $H(\hat{y}, \bar{y}^{ls++})$ . The  $H(\bar{y}^{ls++}, \hat{y})$  loss updates only the parameters in  $G$  and does not affect  $C$  thereby negating the effect of the entropy term  $H(\bar{y}^{ls++})$ . Likewise,  $H(\hat{y}, \bar{y}^{ls++})$  loss updates only matrix  $C$ .

## 4 Experiments

### 4.1 Datasets and Setup

We conducted extensive testing of our approach across a range of datasets, including FashionMNIST [54], SVHN [22], CIFAR10 [24], CIFAR100 [24], FER2013, Animals10N [27], Tiny-ImageNet, and ImageNet-100, employing various network architectures [0, 9, 12, 15, 17, 40]. Due to hardware limitations, we used Tiny-ImageNet and ImageNet-100 as substitutes for the original ImageNet dataset [9]. Tiny-ImageNet features  $64 \times 64$  images with 200 classes, while ImageNet-100 uses 100 classes and the original  $224 \times 224$  image size.

Our methodology was also applied to non-image modalities like Video, Text, and Audio. In the case of the video modality, we utilized UCF101 [28] and HMDB51 [16] datasets, employing Conv+LSTM (CLSTM) and a C3D [30] networks. The Conv+LSTM network utilized a ResNet50 pre-trained on ImageNet as the backbone, with the LSTM layers trained from scratch. The C3D network is a 3D convolution network pre-trained on the Sports-1M dataset [13]. For the text modality, our approach was tested on 20Newsgroup, AGNews [69], and YahooAnswers [39] datasets using pre-trained BERT model [9]. For the Audio modality, we used MelSpectrograms of the GTZAN [51] and SpeechCommands [63] datasets. CNN models pre-trained on ImageNet have shown enhanced generalization on the audio domain [23]. Hence, we trained ResNet50 from scratch and also tested a pre-trained network. Full details of the training augmentations and other details for all the datasets is available in the supplementary material.

Across all tasks, we trained Label Smoothing++ with a consistent setting of  $\alpha = 0.1$ . The matrix  $C$  is stored as pre-softmax values (logits) and was initialized with zeros, resulting in a uniform probability distribution for the non-target classes as the starting point. The code for Label Smoothing++ can be found at <https://github.com/s-choh/LSPP>.

Method	ResNet18	ResNet50	ResNet101	ShuffleNet	DenseNet121
1-hot	64.33	67.47	69.03	60.51	68.15
LS [4]	64.74	67.63	69.30	60.66	68.19
TFKD <sub>self</sub> [5]	-	68.18	-	61.36	68.29
TFKD <sub>reg</sub> [5]	-	68.15	-	60.93	68.37
MBLS [4]	-	65.15	65.81	-	-
Zipf’s LS [13]	59.25	-	-	-	62.64
FL-3 [2]	-	50.31	62.97	-	-
FLSD-53 [2]	-	50.94	62.96	-	-
LS++ (Ours)	<b>65.07±0.08</b>	<b>69.01±0.46</b>	<b>70.04±0.30</b>	<b>63.24±0.49</b>	<b>68.90±0.11</b>

Table 2: Top-1 Classification accuracy on Tiny-ImageNet dataset using different networks.

Dataset	FMNIST	SVHN	CIFAR10	FER	Animals10N	ImageNet-100	
Network	LeNet	LeNet	AlexNet	ResNet18	ResNet18	ResNet18	ResNet50
1hot	82.23±0.34	89.40±0.03	79.98±0.17	70.10±0.21	85.00±0.11	81.72	83.96
LS	82.55±0.62	89.35±0.09	80.66±0.20	70.61±0.10	86.13±0.19	82.22	84.58
TFKD <sub>reg</sub>	82.40±0.26	89.42±0.31	80.78±0.17	<b>70.80±0.41</b>	85.99±0.10	82.44	84.72
OLS	82.97±0.50	89.19±0.43	80.71±0.28	70.67±0.17	86.35±0.38	82.56	84.71
LS++	<b>83.79±0.23</b>	<b>89.77±0.21</b>	<b>81.19±0.05</b>	<b>70.80±0.22</b>	<b>86.51±0.15</b>	<b>82.70</b>	<b>85.06</b>

Table 3: Top-1 Classification accuracy on FashionMNIST (FMNIST), SVHN, CIFAR10, Facial expression recognition (FER), Animals10N, and ImageNet-100 datasets.

Modality	Video				Text			Audio			
	UCF101		HDMB51		20NG AGNews YA			GTZAN		SC	
Network	CLSTM* C3D*		CLSTM* C3D*		BERT*			R50*	R50	R50*	R50
1-hot	71.13	78.56	36.01	45.88	85.02	94.39	77.44	91.50	87.50	96.03	95.13
LS	71.87	81.82	37.91	49.74	85.15	94.50	77.51	92.50	87.50	<b>96.22</b>	95.29
LS++	<b>72.56</b>	<b>82.37</b>	<b>38.24</b>	<b>51.31</b>	<b>85.55</b>	<b>94.67</b>	<b>77.54</b>	<b>93.50</b>	<b>89.00</b>	<b>96.22</b>	<b>95.45</b>

Table 4: Comparison of Top-1 test accuracies on Video, Text, and Audio Modalities. CLSTM: Convolution + LSTM. \* denotes a pre-trained network was finetuned.

## 4.2 Results

We conducted a comprehensive evaluation of Label Smoothing++ (LS++) against other label regularization techniques, including Label Smoothing (LS) [4], Online Label Smoothing (OLS) [5], Margin-based Label Smoothing (MBLS) [4], Teacher-Free Knowledge Distillation (TFKD) [5], and Focal loss [20, 22]. The summarized results can be found in Tables 1, 2, 3, and 4. ‘-’ indicates results were not available in the original paper. Notably, for Tables 3 and 4, baseline experiments were conducted by us using the same setup as ours.

Label Smoothing++ consistently outperformed all compared approaches across different modalities, datasets, and networks. This underscores the superiority of learned mixing probability values over fixed or computed values. Furthermore, we showcase the impact of different methods on the final output probabilities on the FashionMNIST dataset in Figure 4. Figure 4d exposes OLS collapsing to 1-hot training labels. Figure 4b and 4c demonstrate the disruption of inter-class relationships by uniform training labels. Conversely, Figure 4e validates LS++, preserves inter-class relationships, and regularizes outputs.

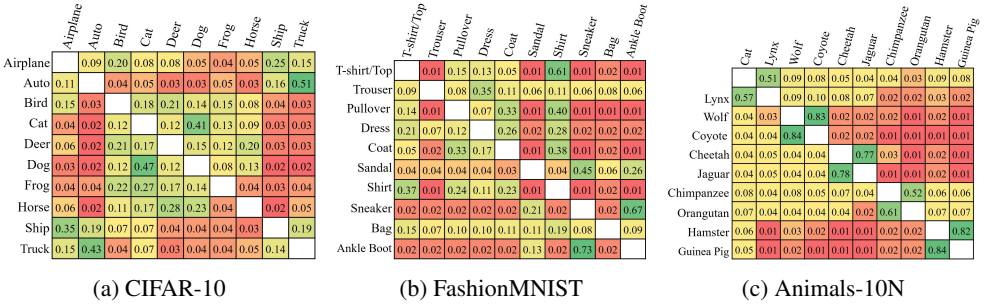


Figure 3: Learned C-Matrices. We can observe that the network favors the semantically close classes while distributing the probabilities and in turn, learns the inter-class relationships.

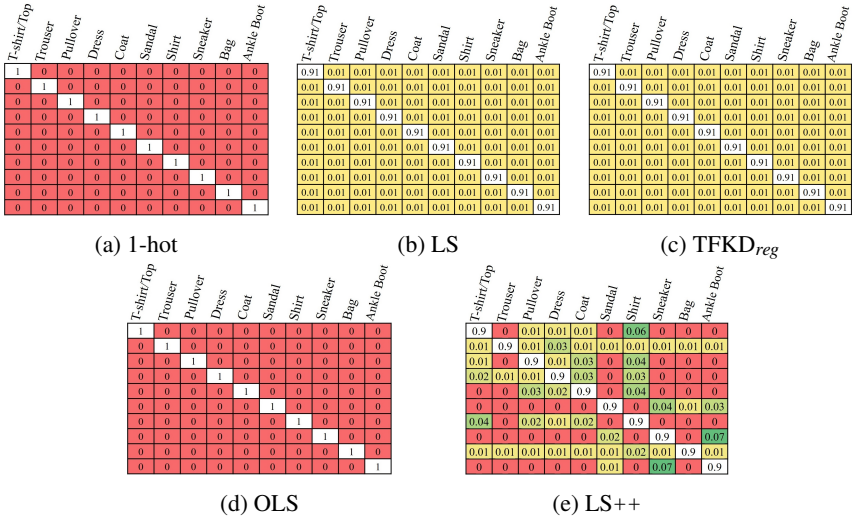


Figure 4: Class-wise output probabilities on the training set of FashionMNIST dataset.

We also present learned C-Matrices on CIFAR10, FashionMNIST, and Animals-10N datasets in Figure 3. The analysis reveals the network’s inclination towards assigning higher probabilities to semantically proximate classes. For instance, in the CIFAR10 dataset, the network exhibits a preference for classes like *Cat* for *Dog*, which are semantically closer. Animals-10N is a fine-grain classification dataset and presents an interesting scenario with 5 pairs of confusing animals. The network consistently assigns probabilities to animals within each pair, considering them as the closest alternatives. We also show C-Matrix for the case of a large number of classes (CIFAR100) in the supplementary material.

## 5 Analysis

### 5.1 Cluster Visualization

In Figure 5, the top row showcases TSNE visualizations [27] of FashionMNIST’s training set. Notably, employing 1-hot targets results in dispersed clusters, while Label Smoothing



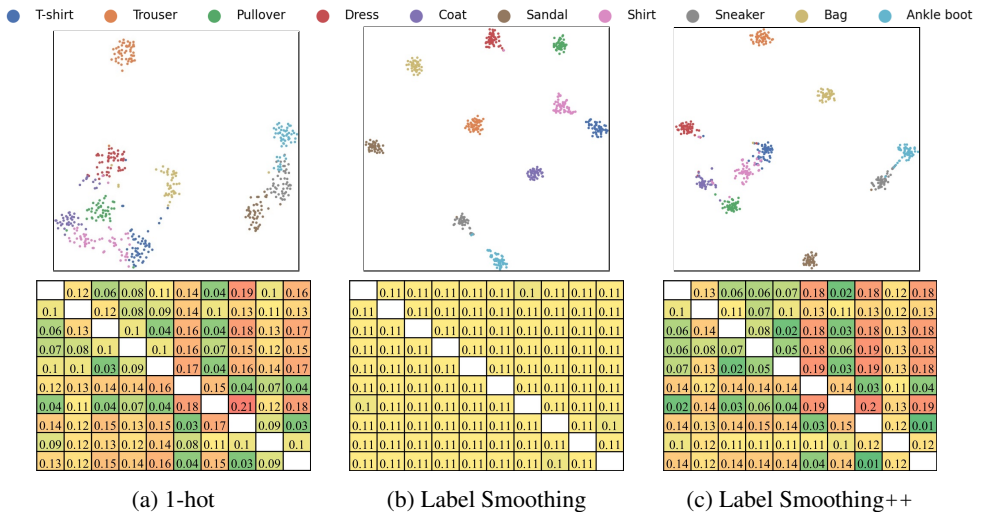


Figure 5: In the upper row, we present TSNE visualizations of various approaches on the FashionMNIST dataset. The lower row shows  $L_1$ -normalized cosine distance among the class cluster centers. 1-hot targets result in dispersed clusters, while Label Smoothing and Label Smoothing++ exhibit more compact clusters. However, Label smoothing places all clusters at an equal distance, effectively eliminating inter-class relationships. In contrast, Label Smoothing++ maintains similar inter-class relationships as observed in 1-hot training.

Method	CIFAR10	CIFAR100	TinyImageNet
Vanilla Cross-Entropy	80.45	78.95	64.93
Symmetric Cross-Entropy-Original	79.92	78.27	64.39
Symmetric Cross-Entropy-Ours	<b>81.19</b>	<b>79.33</b>	<b>65.07</b>

Table 5: Ablation Study on training loss for Label Smoothing++. CE: Cross-entropy, SCE-Original: Original symmetric cross-entropy that updates all parameters, SCE-Ours: Our symmetric cross-entropy loss that updates different parameters.

and Label Smoothing++ yield more compact ones. Compact clusters are pivotal in minimizing collisions and enhancing generalization capabilities. In the bottom row of Figure 5, we analyze the  $L_1$ -normalized cosine distance among class cluster centers. Label smoothing evenly spaces out all clusters, effectively eliminating inter-class relationships. On the other hand, Label Smoothing++ has the optimal effect that generates compact clusters, reduces overconfidence, and achieves high generalization while preserving inter-class relationships.

## 5.2 Ablation Study

We perform an ablation study focusing on selecting training loss for Label Smoothing++. Our investigation highlights distinctions in outcomes and the acquired  $C$  matrix through various loss functions, including standard cross-entropy (CE), original symmetric cross-entropy (SCE-Original), and our symmetric cross-entropy (SCE-Ours). The matrix corresponding to CIFAR10 can be found in Figure 6. Notably, both cross-entropy and original symmetric



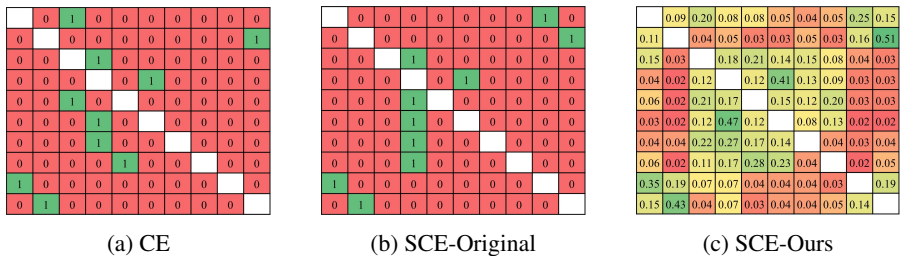


Figure 6:  $C$ -Matrix learned by cross-entropy, symmetric cross-entropy (SCE-original), and symmetric cross-entropy (SCE-ours) on CIFAR10. Cross-entropy and original symmetric cross-entropy result in a low entropy matrix and only our loss function provides the desired results. The targets can be generated by mixing these matrices with a 1-hot vector as per  $\alpha$ .

Dataset	CIFAR100				Tiny-ImageNet		
	R34→R18	R34→R34	R34→R50	R50→SN	R50→SN	R101→SN	D121→SN
1-hot	78.67	79.09	80.83	72.51	66.56	66.39	66.39
LS	79.40	80.15	81.15	71.70	66.58	66.74	66.80
LS++	<b>79.90</b>	<b>80.38</b>	<b>81.16</b>	<b>72.64</b>	<b>66.89</b>	<b>67.00</b>	<b>67.43</b>
PT-LS++	79.75	79.90	80.93	72.59	64.10	64.76	64.75

Table 6: Comparison of various teachers in the context of knowledge distillation, we consider three different teacher models: 1-hot, Label Smoothing (LS), and Label Smoothing++ (LS++), each trained with their respective loss functions. Additionally, we introduce a proxy teacher model (PT-LS++), where the  $C$ -Matrix learned by the teacher network (trained with LS++) serves as the guiding information for training the student models.

cross-entropy result in low entropy vectors. This outcome stems from the indirect entropy minimization loss, as elaborated in section 3.2. Contrastingly, our symmetric cross-entropy achieves the desired result by appropriately distributing probabilities. The impact of this on generalization is illustrated in Table 5, revealing a noticeable degradation in performance.

### 5.3 Knowledge Distillation with a Proxy Teacher

In knowledge distillation, a teacher network plays a guiding role in training a student network. The teacher network understands inter-class relations within a sample across all classes and generates regularized training labels for the student, enhancing its generalization. In our scenario, the  $C$ -Matrix serves as a proxy teacher (PT-LS++) in the absence of a teacher network. Leveraging the  $C$ -Matrix allows us to generate regularized training labels for each class instead of per sample. We conducted experiments on CIFAR-100 and Tiny-ImageNets for various transfer tasks. The student networks were trained using traditional cross-entropy loss, except that a teacher network provided training labels. For the proxy teacher (PT-LS++), training labels were created using the learned  $C$ -Matrix of the teacher network (trained with Label Smoothing++).

The results are in Table 6. As expected, knowledge distillation improves accuracy in all cases, with the network trained with LS++ acting as the most effective teacher. The proxy teacher (PT-LS++) achieves lower performance compared to other teachers but still outper-

Dataset	CIFAR100	FashionMNIST	TinyImageNet	ImageNet-100
1-hot	79.38	86.66	64.33	81.72
LS++	80.25 <sub>↑</sub>	87.47 <sub>↑</sub>	65.07 <sub>↑</sub>	82.22 <sub>↑</sub>
Co [B]	80.11	88.36	65.86	82.86
Co + LS++	80.44 <sub>↑</sub>	88.92 <sub>↑</sub>	66.53 <sub>↑</sub>	83.04 <sub>↑</sub>
Mx [BS]	81.31	88.48	66.17	81.88
Mx + LS++	81.46 <sub>↑</sub>	88.59 <sub>↑</sub>	66.41 <sub>↑</sub>	82.88 <sub>↑</sub>
Cx [BS]	81.95	88.11	68.50	83.50
Cx + LS++	<b>82.24</b> <sub>↑</sub>	88.27 <sub>↑</sub>	<b>68.62</b> <sub>↑</sub>	<b>83.66</b> <sub>↑</sub>
RA [B]	80.01	92.40	65.87	82.88
RA + LS++	80.24 <sub>↑</sub>	<b>92.85</b> <sub>↑</sub>	66.05 <sub>↑</sub>	83.54 <sub>↑</sub>

Table 7: Application of Label Smoothing++ with Input Augmentations techniques - Co: Cutout, Mx: Mixup, Cx: CutMix, RA: RandAugment.

forms training the network directly (refer Table 1 and 2). The biggest advantage of the proxy teacher is its independence from the teacher model’s output, which can be computationally expensive. In our ResNet101 → ShuffleNet experiments on TinyImageNet, the proxy teacher took only half the time to train the student compared to traditional knowledge distillation.

## 5.4 Compatibility with Input Augmentations

In this section, we assess the compatibility of Label Smoothing++ with input augmentation techniques such as Cutout, Mixup, Cutmix, and Randaugment. The results of this experiment are presented in Table 7 using CIFAR100, FashionMNIST, Tiny-ImageNet, and ImageNet-100 datasets with ResNet34, ResNet18, ResNet18, and ResNet18, respectively. Our findings indicate that label regularization seamlessly integrates with input regularization techniques. Employing input and label regularization together yields optimal performance, as evidenced by the results in the table.

## 6 Conclusion

In this paper, we introduced a label regularization technique termed Label Smoothing++, designed to enable neural networks to select their optimal training labels. Our approach uses different training labels for each class while ensuring that samples within the same class yield consistent outputs. The training labels collectively form a  $C$ -Matrix which captures the inter-class relationships and serves as a proxy teacher for knowledge distillation. Our proposed label regularization approach is compatible with input regularization and provides a performance boost when used together. Extensive experimentation across various datasets demonstrates that Label Smoothing++ reduces overconfidence and promotes high generalization and inter-class relationships.

**Acknowledgements** This work was supported in part by the following grants: National Institutes of Health Grant RF1AG073424, National Institutes of Health Grant P30AG072980, and Arizona Department of Health Services Grant CTR057001. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

## References

- [1] Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Patchswap: A regularization technique for vision transformers. In *BMVC*, page 996, 2022.
- [2] Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Generative alignment of posterior probabilities for source-free domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4125–4134, 2023.
- [3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL <http://arxiv.org/abs/1810.04805>.
- [6] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017. URL <http://arxiv.org/abs/1708.04552>.
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.
- [8] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1055–1064, 2021.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [10] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL <https://openreview.net/forum?id=SlgmrXHFvB>.
- [11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

- [13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [16] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.
- [17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [18] Jiajun Liang, Linze Li, Zhaodong Bing, Borui Zhao, Yao Tang, Bo Lin, and Haoqiang Fan. Efficient one pass self-distillation with zipf’s label smoothing. In *European conference on computer vision*, pages 104–119. Springer, 2022.
- [19] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020.
- [20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [21] Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 80–88, 2022.
- [22] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.
- [23] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.
- [24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [25] Kamallesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking CNN models for audio classification. *CoRR*, abs/2007.11154, 2020. URL <https://arxiv.org/abs/2007.11154>.

- [26] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=HyhbYrGYe>.
- [27] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019.
- [28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. URL <http://arxiv.org/abs/1212.0402>.
- [29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [31] George Tzanetakis. Automatic musical genre classification of audio signals. In *ISMIR 2001, 2nd International Symposium on Music Information Retrieval, Indiana University, Bloomington, Indiana, USA, October 15-17, 2001, Proceedings*, 2001. URL <http://ismir2001.ismir.net/pdf/tzanetakis.pdf>.
- [32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [33] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR*, abs/1804.03209, 2018. URL <http://arxiv.org/abs/1804.03209>.
- [34] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.
- [35] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.
- [36] Sangdoon Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6022–6031. IEEE, 2019. doi: 10.1109/ICCV.2019.00612. URL <https://doi.org/10.1109/ICCV.2019.00612>.
- [37] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996, 2021.

- [38] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=r1Ddp1-Rb>.
- [39] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.
- [40] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.